Learning Dictionary of Discriminative Part Detectors for Image Categorization and Cosegmentation

Jian Sun, *Member, IEEE,* Jean Ponce, *Fellow, IEEE,*

Abstract

This paper proposes a novel approach to learning mid-level image models for image categorization and cosegmentation. We represent each image class by a dictionary of discriminative part detectors that best discriminate that class from the background. We learn category-specific part detectors in a weakly supervised setting in which the training images are only labeled with category labels without part / object location labels. We use a latent SVM model regularized by $l_{1,2}$ group sparsity to learn the discriminative part detectors. Starting from a large set of initial parts, the group sparsity regularizer forces the model to jointly select and optimize a set of discriminative part detectors in a max-margin framework. We propose a stochastic version of a proximal algorithm to solve the corresponding optimization problem. We apply the learned part detectors to image classification and cosegmentation, and quantitative experiments with standard benchmarks show that our approach matches or improves upon the state of the art.

Index Terms

Part detector, image classification, image cosegmentation, group sparsity

1 INTRODUCTION

Learning mid-level image representations is a promising approach to improving the performance of image recognition systems. Traditional recognition systems model the set of low-level features (e.g.,

- *This work was done when Jian Sun was working as a postdoctoral researcher in WILLOW project-team, Département d'Informatique de l'Ecole Normale Supérieure, ENS/INRIA/CNRS UMR 8548.*
- *Jian Sun is with the School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, Shaanxi, 710049, P. R. China. E-mail: jiansun@mail.xjtu.edu.cn.*
- *Jean Ponce is with Département d'Informatique de l'Ecole Normale Supérieure, 45 Rue d'Ulm 75005, Paris, France. Email: jean.ponce@ens.fr.*

SIFT [1], HOG [2]) by a mid-level bag-of-words model [3], sparse codes [4], Fisher vectors [5], etc. These approaches generally represent an image by a fixed-length image code through quantitizing the low-level feature space, then feed these image codes to classifiers for image recognition. They have been shown to be effective for image recognition.

Another category of popular mid-level representation decomposes objects, scenes, or images into parts [6], [7], [8], [9], [10], [11], and each part covers a discriminative region of an object / image, e.g., the head of dogs, the rear of cars. Successful examples of part-based models include the deformable part models (DPMs) [9], poselets [7], discriminative patches [11], [10], [12] for object detection [7], [9], action recognition [13], semantic segmentation [6], scene classification [11], [10], [12], etc.

(a) Example images of “Car” category (b) Examples of learned part detectors

Fig. 1. We learn discriminative part detectors for an image set with the same category label. The part detectors are applied to image classification and cosegmentation. (*Best viewed in color.*)

Learning part-based models has, however, been a challenge. The essential question is how to efficiently learn and select object / image parts that are discriminative for an image / object category. The deformable part model (DPM) [9] learns a mixture of object templates in different poses represented by a few spatially deformable object parts using a discriminative latent-SVM learning framework. The positions and number of parts are heuristically initialized given the object bounding box. Other recent methods learn a much larger set of discriminative part detectors. For example, in poselet [7] and discriminative patch (DP) [8], [11], [12] models, a large number of part detectors are first learned by linear SVMs from image patch clusters. Discriminative parts are then selected by ranking the importance of image parts and discarding the unimportant ones. In the case of poselets, additional supervision in the form of keypoint labels is necessary.

In this work, we propose a principled approach to learning class-specific part detectors inspired by dictionary learning approaches [14], [15]. As illustrated in Figure 1, given a set of training images (Figure 1 (a)) from the same category, we design a novel latent SVM model regularized by group sparsity to jointly select and optimize a set of discriminative part detectors. Given a large set of initial parts, the group sparsity regularizer forces the model to automatically select and optimize a dictionary of discriminative part detectors in a max-margin framework. Our model tends to select the discriminative part detectors that more frequently and strongly appear in positive training images than in the negative ones. Examples of the learned part detectors are shown in Figure 1 (b).

With our approach, part detectors are learned to reliably detect the discriminative image parts that can best discriminate the category of interest from the background world. We have applied the learned part detectors to image classification and cosegmentation. For image classification, we encode an image using a fixed-length mid-level code by max-pooling the responses of the learned part detectors to the image, and achieve competitive performance in the applications of object / scene / event classification over benchmark databases. We have observed that our discriminative part detectors are able to detect the common object parts from a set of images containing the same object class, and therefore propose a novel cosegmentation model in a discriminative clustering framework by incorporating the object cues provided by the learned part detectors. We also report state-of-the-art results on benchmark datasets.

A preliminary version of this work appeared in [16]. In this journal version, we extended the conference version in [16] as follows. First, we present more implementation details on algorithms and experiments. Second, we re-implement the algorithm of learning part detectors using training images in multi-scale pyramids, and accordingly report our improved classification and cosegmentation results. We also test the effect of the different part initialization methods on the recognition performance. The source codes of our algorithm are also published on the link of *https://github.com/exploreman/discriminative_parts*.

1.1 Related Work

1.1.1 Image Representation

Traditional image representations are primarily based on quantization of low-level features, e.g., bag-of-words (BoWs) [3], sparse coding [4], Fisher vector [5], LLC coding [17], etc. The image is represented by spatially pooling the corresponding codes globally on a coarse grid or a spatial pyramid [18] for image classification. These approaches have achieved excellent results for image recognition. Contrary to these approaches, we learn a dictionary of discriminative mid-level image parts in diverse poses / viewpoints, which directly represent object or image category by their mid-level parts.

There is a large body of work on part-based models for recognition. The deformable part model (DPM) [9] represents an object by a set of deformable parts organized in a tree structure and learned from object bounding boxes. Strongly-supervised DPM [19] further incorporates human-annotated object

parts to improve performance. In poselets [7], a large number of object parts are learned and selected using SVMs trained over clusters of image patches with the aid of human-labelled 3D keypoints in different poses. Discriminative patch (DP) methods learn distinctive image patches using discriminative clustering [11] or extended mean-shift mode seeking [12]. Both the poselet and DP methods separately learn a set of part detectors using linear SVMs and select the distinctive ones by heuristically ranking their importance.

Contrary to these approaches, we propose a unified model to jointly learn and select a dictionary of category-specific part detectors using a latent-svm model with group sparsity regularization. Our approach works in a weakly supervised way and only requires the training examples at the category level without any manually labelled keypoints or parts. The group sparsity regularizer plays the role of part selector, and allows us to select diverse and discriminative part detectors best discriminating the positive training examples from negative background. The number of discrimiantive part detectors can be controlled by the group sparsity regularization coefficient.

Our approach is related to dictionary learning approaches [20], [15], [21], [22], where image patches are encoded as a sparse linear combination of basis (dictionary elements) optimized for image reconstruction [15], [22] or classification [21], [20]. Our part learning model bears similarities to the dictionary learning approaches but is significantly different. Our learned part detectors are similar to the basis used in dictionary learning, but they are specifically optimized for object / image part detection, which requires a novel latent SVM model with group sparsity regularization for learning the dictionary of part detectors.

1.1.2 Cosegmentation

Cosegmentation [23], [24], [25] is the problem of jointly segmenting a set of images into foreground and background regions. It is a challenging task in computer vision, since it involves a weak form of supervision, i.e., images contain instances of the same object class, to segment out these objects. Its multi-class extensions [26], [27] try to segment out multiple classes of objects from images. Recently, discriminative clustering [28] has successfully been applied to image cosegmentation and achieved state-of-the-art cosegmentation results. In this paper, we address the two-class cosegmentation problem to segment out common objects from diverse backgrounds. We take the image set containing the same object as positive training data, and the external background images as negative training data, our approach can learn a dictionary of object part detectors which are discriminative and frequently appear in the positive training images. These part detectors provide object localization cues for better object cosegmentation.

The rest of our paper is organized as follows. Section 2 formally defines our part model. Section 3 presents our model for learning discriminative part detectors. Section 4 discusses our approach to solving the corresponding optimization problem. Applications of discriminative part detectors to image classification and cosegmentation are presented in Section 5. Section 6 experimentally illustrate the

effectiveness of the proposed approach on benchmark databases. This paper concludes with a brief discussion in section 7.

2 Part Detector Definition

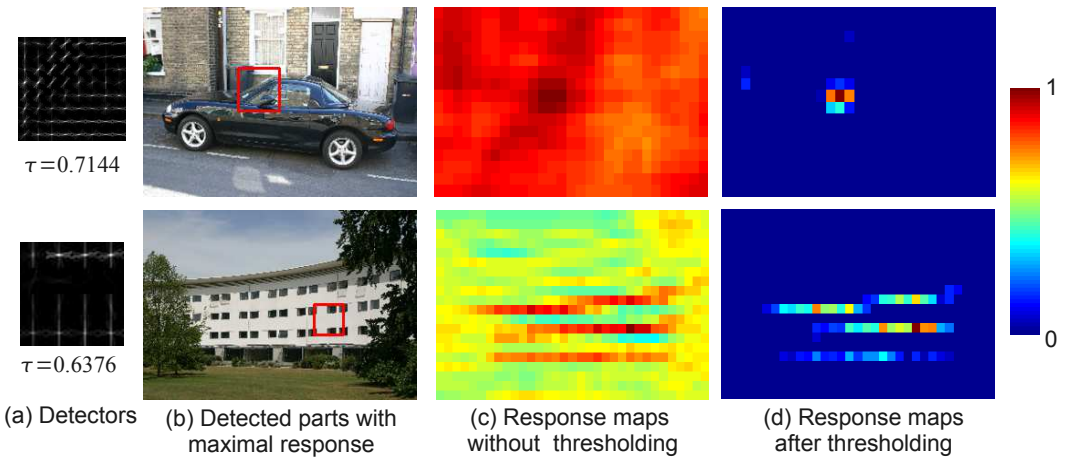


Fig. 2. Examples of part detectors. With the learned part thresholds, part detectors can produce clean responses to images. (*Best viewed in color.*)

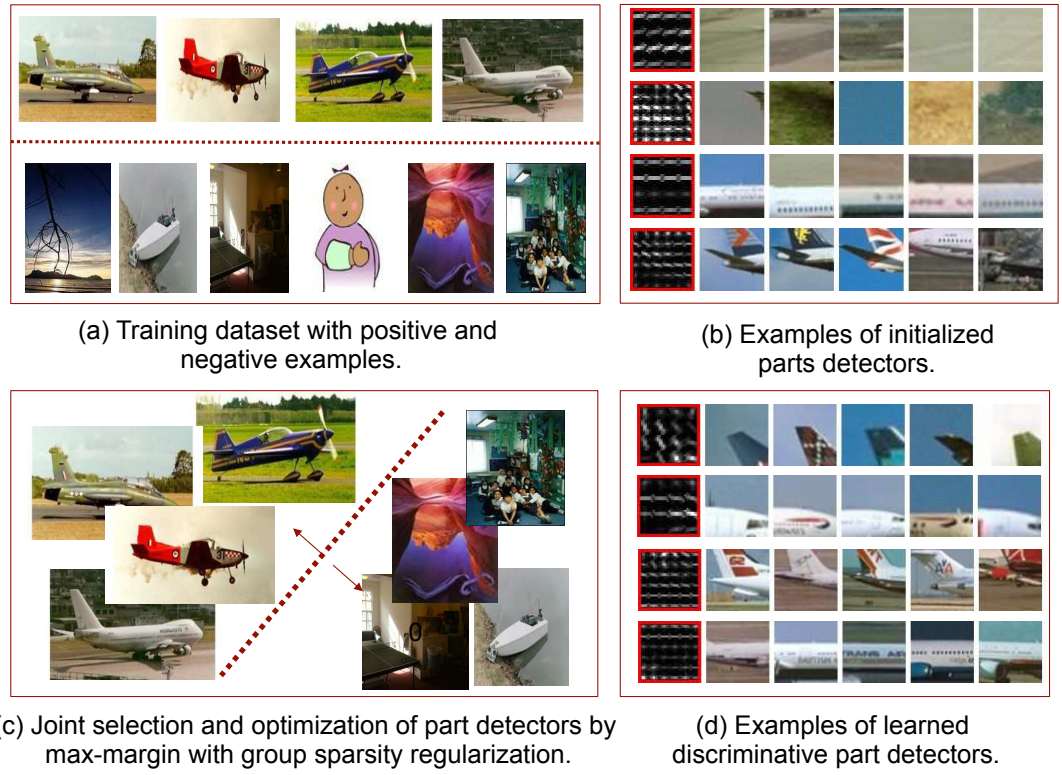
Given an image I, let us consider dense features extracted at fixed intervals over the image grid. An *image part* is a box whose top-left corner is positioned at z, and it is represented by a feature vector $\Phi(I, z)$ that concatenates all the feature vectors within the box. We further define a *part detector* $\Gamma_k = (\beta_k, \tau_k)$ $(k = 1, \cdots, K)$ as *template / threshold* pair $(\beta_k,\ \tau_k)$, and define its response to image part $\Phi(I, z)$ as

$$r_z(\Gamma_k, I) = [S(\beta_k, \Phi(I, z)) - \tau_k]_+, \tag{1}$$

where $[a]_+ = \max(a, 0)$, and $S(\beta_k, \Phi(I, z))$ is the *matching score* between the part template β_k and the image part $\Phi(I, z)$. In this work, we simply define the matching score as the inner product between part template and normalized part feature vector:

$$S(\beta_k, \Phi(I, z)) \quad = \quad \frac{< \beta_k, \Phi(I, z) >}{||\Phi(I, z)||_2} =< \beta_k, \frac{\Phi(I, z)}{||\Phi(I, z)||_2} > \tag{2}$$

Based on Eq.(1), the part detector Γ_k has non-zero response to image I at position z only when the matching score $S(\beta_k, \Phi(I, z))$ is higher than τ_k. Furthermore, we say that the part Γ_k *appears in an image* I when there exists at least one position z that satisfies $r_z(\Gamma_k, I) > 0$. Figure 2 shows examples of part detectors and the corresponding responses. As shown in this figure, after thresholding the matching scores using Eq.(1), irrelevant image parts are suppressed and only significantly similar image parts have non-zero responses.



(a) Training dataset with positive and negative examples.

(b) Examples of initialized parts detectors.

(c) Joint selection and optimization of part detectors by max-margin with group sparsity regularization.

(d) Examples of learned discriminative part detectors.

Fig. 3. An illustration of our learning framework. Given a training set of positive and negative images for an image category, we first initialize a set of part detectors as discussed in Section 3.1. Then we jointly select and optimize a set of part detectors (i.e., template / threshold pairs) using the novel latent SVM model regularized by group sparsity as discussed in Section 3.2.

3 Learning Part Detectors by Group Sparsity

In this section, we aim to learn a set of category-specific image part detectors that can best discriminate the images in the category of interest from the background images. As shown by Figure 3, the input of our approach is an image set composed of positive and negative training examples. First, we automatically pick an initial set of candidate part detectors associated with the image category. They frequently appear in the positive training images but may not be discriminative. Then we use a novel latent SVM model to select and optimize discriminative part detectors with group sparsity regularization.

3.1 Part Detectors Initialization

We first initialize a set of part detectors which will be taken as candidates for futher optimization and selection by our learning model. We have tried two types of initialization methods, compared experimentally in Section 7.2.2.

Random Sampling. To initialize the candidate part detectors for an image category, we randomly crop a fixed number of image parts from the positive training images. Assume that we have K sampled image parts, then we initialize K part detectors $\{\Gamma_k\}_{k=1}^K$, $\Gamma_k = \{\beta_k, \tau_k\}$. Each part template β_k is taken as the feature vector of the k-th random patch, and the part threshold τ_k is initially set to zero value.

Patch Clustering. An alternative initialization approach is based on patch clustering. We first randomly crop a large number of image parts (approximately ten thousands) from the positive training images, then we perform K-means clustering ($K = 1000$ clusters in our implementation) over these sampled image parts. This is similar to the construction of a visual word dictionary in BoWs. We only retain sufficiently large clusters of size 10 or more. Assume that we have K clusters of image parts, then we initialize K part detectors $\{\Gamma_k\}_{k=1}^K$. The part template β_k and part threshold τ_k are initialized with the k-th cluster center and a zero value respectively.

3.2 Learning Discriminative Part Detectors

With the above initialization, we now learn a set of part detectors that best discriminate the positive and negative training images. We require that the learned part detectors should appear more frequently and strongly in the positive training images than in the negative ones.

Before introducing our learning method, let us first define the confidence of image I belonging to the current category given class-specific part detectors $\Gamma = \{\Gamma_k\}_{k=1}^K$:

$$g(I, \Gamma) = \sum_{k=1}^{K} [\beta_k^T \Phi(I, z_k) - \tau_k]_+, \tag{3}$$

where z_k is a latent variable indicating the image part position with maximum response:

$$z_k = \text{argmax}_{z \in \Omega_I} \, \beta_k^T \Phi(I, z), \tag{4}$$

and Ω_I defines the set of all possible part positions in I. Observe from Eq.(3) that $g(I, \Gamma) \geq 0$ is defined as the sum of the maximum responses of all the part detectors to image I. Image I thus has higher confidence in belonging to the category of interest when more parts appear in I and have higher responses.

Next, we learn part detectors using a variant of the latent SVM model with group sparsity regularization. The basic idea is to jointly select and optimize the part detectors by maximizing the margin of the confidence value $g(I, \Gamma)$ on positive and negative training images. Denote the training image set as $\{I_n, y_n\}_{n=1}^N$ where $y_n = 1$ if I_n belongs to the category and $y_n = -1$ otherwise. The cost function is defined as:

$$E(\Gamma, b) = \frac{1}{N} \sum_{n=1}^{N} L(g(I_n, \Gamma), y_n, b) + \lambda R(B), \tag{5}$$

where $B = \{\beta_k\}_{k=1}^K$ is the set of all part templates and L is the squared hinge loss function:

$$L\left(g(I, \Gamma), y, b\right) = [1 - y(g(I, \Gamma) + b)]_+^2, \tag{6}$$

and b is a bias term. We have chosen this loss function because it is differentiable w.r.t. g and b. We could have used other differentiable losses, e.g., a logistic function.

$R(B)$ is a regularization term over the part templates. We impose group sparsity [29] over part templates, where each template is considered as a group. This forces the algorithm to automatically select a few discriminative part detectors with non-zero templates from a large set of candidate part detectors. Typical group sparsity terms include $l_{1,2}$ and $l_{1,\infty}$ regularizers [29]. We choose the $l_{1,2}$ structured sparsity norm in this paper, i.e., $R(B) = \sum_{k=1}^{K} ||\beta_k||_2$, which is the sum of l_2 norm of part templates, and is convex w.r.t. B. In summary, we learn the discriminative part detectors by solving:

$$\underset{\Gamma,b}{\operatorname{argmin}} \left\{ \frac{1}{N} \sum_{n=1}^{N} [1 - y_n(g(I_n, \Gamma) + b)]_+^2 + \lambda \sum_{k=1}^{K} ||\beta_k||_2 \right\}, \tag{7}$$

where $g(I_n, \Gamma)$ depends on latent variables in Eq.(4).

The above variant of the latent SVM model tries to enforce that $g(I, \Gamma) + b \geq 1$ if I is positive training image, and $g(I, \Gamma) + b \leq -1$ if I is negative training image. This forces the learned part detectors to have larger responses to positive training images than to negative ones. It implies that the learned part detectors should be *discriminative*, i.e., more frequently and strongly trigger in the positive training images than in the negative ones. With group sparsity regularization, the optimization procedure will automatically discard the less discriminative part detectors among the initial ones.

Let us briefly compare our model to the latent SVM in [9]. Using the squared hinge loss instead of the regular one is a minor difference. More importantly, our proposed latent SVM model is regularized by group sparsity, which is able to automatically select discriminative part detectors from a large pool of initial detectors. Second, our learned part detectors are template and threshold pairs. With the part thresholds, parts are not required to appear in every image of the category, which makes the detectors robust to intra-class variations caused by poses, sub-categories, etc.

4 Optimization Algorithm

The latent model of Eq.(7) is semi-convex [9] w.r.t. the part detectors Γ, i.e., it is convex for the negative examples and non-convex for the positive examples. This can be justified by the following facts. First, $g(I, \Gamma)$ is convex w.r.t. $\Gamma = \{\beta_k, \tau_k\}_{k=1}^{K}$. This can be easily shown by noting that $g(I, \Gamma) = \sum_{k=1}^{K} \max\{\tilde{\beta}_k^T \tilde{\Phi}(I, z_k), 0\}$, where $\tilde{\beta}_k = [\beta_k^T, \tau_k]^T$ and $\tilde{\Phi}(I, z_k) = [\Phi^T(I, z_k), -1]^T$, which is the maximum of linear functions. Second, the cost function in Eq.(7) is convex and non-decreasing w.r.t. $g(I, \Gamma)$ if I is a negative example (i.e., $y = -1$). Therefore the cost is convex w.r.t. Γ for the negative examples. However, it is non-convex for the positive examples.

Following [9], we optimize Eq. (7) by iteratively performing the following two steps. First, we update the latent variables for all the positive examples based on Eq. (4). Second, given the set of latent variables for all the positive examples (denoted as Z_p), we optimize part detectors $\{\beta_k, \tau_k\}_{k=1}^{K}$ and bias term b by

minimizing the convex cost $E(\Gamma, b; Z_p)$ which is the cost function in Eq.(7) with fixed latent variables for positive examples. We stop the iterations when a maximal number of steps is reached or when the parameters do not change significantly any more.

We now discuss how to minimize $E(\Gamma, b; Z_p)$ given Z_p. This cost function is smooth for b and piecewise-smooth for Γ. Therefore, we utilize a gradient descent method to optimize b and a subgradient method to optimize $\Gamma = \{\beta_k, \tau_k\}_{k=1}^K$ simultaneously. Due to the group sparsity regularization for $\{\beta_k\}_{k=1}^K$, we utilize a stochastic version of a proximal method (specifically, the FISTA algorithm [30]) for the optimization of part detectors by minimizing the convex cost $E(\Gamma, b; Z_p)$. Proximal methods are known to be effective in optimizing convex loss functions with sparse regularization. For an objective function with the form of $\min_B\{L(B) + \lambda R(B)\}$ where L is an convex loss function and $R(B)$ is the above defined group sparsity regularization over $B = \{\beta_k\}_{k=1}^K$, it can be efficiently optimized by updating the parameters using a proximal operator [30]:

$$\beta_k{}^{t+1} = \mathrm{Prox}_{\lambda\gamma}(\beta_k^t - \gamma \frac{\partial L(B)}{\partial \beta_k{}^t}), \tag{8}$$

where

$$\mathrm{Prox}_{\mu}(\boldsymbol{\beta_k}) = \frac{1}{||\boldsymbol{\beta_k}||_2}\boldsymbol{\beta_k}[||\boldsymbol{\beta_k}||_2 - \mu]_+ \tag{9}$$

for $l_{1,2}$ group sparsity regularizer.

In summary, given training image set, we minimize the energy $E(\Gamma, b; Z_p)$ by iteratively updating the parameters:

$$\beta_k{}^{t+1} = \mathrm{Prox}_{\lambda\gamma}(\beta_k^t - \gamma \frac{1}{N}\sum_{n=1}^{N} \frac{\partial L_n}{\partial \beta_k{}^t}), \tag{10}$$

$$b^{t+1} = b^t - \gamma \frac{1}{N}\sum_{n=1}^{N} \frac{\partial L_n}{\partial b^t}, \tag{11}$$

$$\tau_k{}^{t+1} = \tau_k^t - \gamma \frac{1}{N}\sum_{n=1}^{N} \frac{\partial L_n}{\partial \tau_k}, \tag{12}$$

where γ is the step size determined by the back-tracking method in the FISTA algorithm [30], and $L_n = L(g(I_n, \Gamma), y_n, b)$. The gradient (w.r.t. b) and sub-gradients (w.r.t. β_k, τ_k) involved are computed as follows.

$$\frac{\partial L_n}{\partial b} = \begin{cases} -\eta_n y_n & \text{if } y_n(g(I_n, \Gamma) + b) < 1 \\ 0 & \text{otherwise,} \end{cases} \tag{13}$$

$$\frac{\partial L_n}{\partial \beta_k} = \begin{cases} -\eta_n y_n \Phi(I_n, z_{n,k}) & \text{if } \mathbf{C} \text{ is satisfied} \\ 0 & \text{otherwise,} \end{cases} \tag{14}$$

$$\frac{\partial L_n}{\partial \tau_k} = \begin{cases} \eta_n y_n & \text{if } \mathbf{C} \text{ is satisfied} \\ 0 & \text{otherwise,} \end{cases} \tag{15}$$

Algorithm 1 Algorithm for discriminative learning of class-specific part detectors.

Input: Training images $\mathbb{S} = \{I_n, y_n\}_{n=1}^N$. Maximum iterations T_{in} and T_{out}.

Output: Learned part detectors $\Theta = \{\beta_k, \tau_k\}_{k=1}^K$.

Initialize part detectors $\Gamma^0 = \{\beta_k^0, \tau_k^0\}_{k=1}^K$ as in Section 3.1, bias term $b = 0$ and $t_{out} = 0$;
while $t_{out} < T_{out}$ **do**
  Compute latent variables of all part detectors over positive training images by Eq.(4), then optimize part detections by the following FISTA iterations.
  Initialize $s_1 = s_0 = 1; \overline{\Theta}^1 = \overline{\Theta}^0 = \Theta^0 = \Gamma^{t_{out}}$; $t = 0$.
  while $t < T_{in}$ **do**
    Sample training examples $\mathbb{S}_t \subset \mathbb{S}$ (six positive and negative examples respectively).
    Compute latent variables for part detectors over sampled negative examples by Eq.(4).
    Compute the estimated average gradients of parameters (denoted as $\frac{\partial \hat{L}}{\partial \beta_k^t}$, $\frac{\partial \hat{L}}{\partial \tau_k}$, $\frac{\partial \hat{L}}{\partial b}$) using Eqs. (13-15) over S_t; Estimate Lipschitz constant γ^t by backtracking as in the FISTA algorithm [30];
    Update parameters: $\overline{\beta}_k^t = \mathrm{Prox}_{\lambda/\gamma_t}(\beta_k^t - \frac{1}{\gamma_t}\frac{\partial \hat{L}}{\partial \beta_k^t})$; $\overline{\tau}_k^t = \tau_k^t - \frac{1}{\gamma_t}\frac{\partial \hat{L}}{\partial \tau_k}$; $\overline{b}^t = b^t - \frac{1}{\gamma_t}\frac{\partial \hat{L}}{\partial b}$, for $k = 1, \cdots K$. Then assign $\overline{\Theta}^t \leftarrow \{\overline{\beta}_k^t, \overline{\tau}_k^t\}_{k=1}^K$;
    $s_{t+1} = \frac{1+\sqrt{1+4s_t^2}}{2}$;
    $\Theta^{t+1} = \overline{\Theta}^t + \frac{s_t - 1}{s_{t+1}}(\overline{\Theta}^t - \overline{\Theta}^{t-1})$;
    $t = t + 1$.
  end while
  $\Gamma^{t_{out}} = \Theta^{T_{in}}$; $t_{out} = t_{out} + 1$.
end while
Output the learned part detectors set Θ which is composed of the part detectors in $\Gamma^{t_{out}}$ with non-zero norms in the part templates.

where $\eta_n = 2(1 - y_n(g(I_n, \Gamma) + b))$, $z_{n,k}$ is the k-th latent variable for image I_n, **C** denotes the conditions of $\beta_k^T \Phi(I_n, z_{n,k}) > \tau_k$ and $y_n(g(I_n, \Gamma) + b) < 1$. The optimization of $E(\Gamma, b; Z_p)$ is a large-scale and high-dimensional convex optimization problem. To make it tractable, we propose to use a stochastic algorithm in which a subset (six random samples) of training images are sampled to approximate the gradients / subgradients [31].

Algorithm 1 presents the detailed optimization procedures. After optimization, non-discriminative part templates are set to zero due to the $l_{1,2}$ regularization. We discard these part detectors with zero part templates and derive a set of discriminative part detectors.

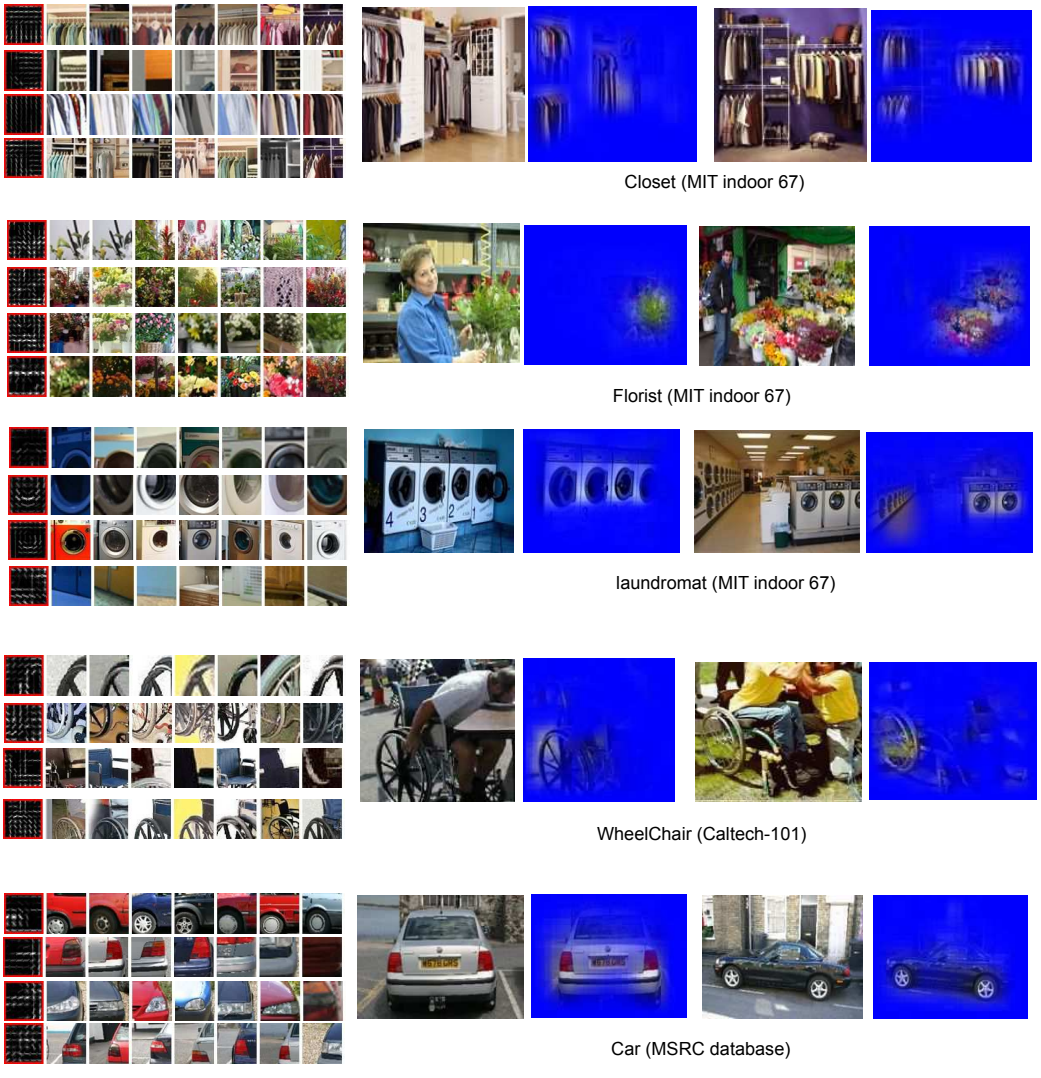


Fig. 4. Examples of learned part detectors, detected parts and total response maps of part detectors to images. The learned part detectors have higher responses to the discriminative regions in each category. Response maps are shown as the original images masked by the linearly normalized total response maps in range of [0, 1]. (*Best viewed in color.*)

5 TOTAL RESPONSE MAPS OF PART DETECTORS

To illustrate the learned part detectors, we define the *response map of a part detector* Γ_k to an image I as the weighted sum of all the detected parts appearing in the image pyramid by resizing the image to multi-scale resolutions, i.e.,

$$R(\Gamma_k, I) = \sum_s \sum_{z \in \Omega_{I^s}} r_z(\Gamma_k, I^s) M_z(I^s), \tag{16}$$

where I^s is the image at scale s, $r_z(\Gamma_k, I^s)$ is the response value defined in Eq.(1), $M_z(I^s)$ is the binary mask of I^s indicating the region occupied by image part located at position z. The part mask $M_z(I^s)$ is rescaled by $\frac{1}{s}$, therefore the response map $R(\Gamma_k, I)$ has the same resolution as I. In our implementation, we construct an image pyramid using thirteen scaling factors, i.e., $s \in \{2^{-2}, 2^{-1.75}, \cdots, 2^{0.75}, 2\}$. The *total response map* to an image is defined as the sum of all the response maps of the derived part detectors:

$$R(\Gamma, I) = \sum_k R(\Gamma_k, I). \tag{17}$$

Figure 4 shows examples of learned part detectors and detected parts. As shown in Figure 4(a), the learned detectors are discriminative for the categories considered. For example, in categories of closet, florist, laundarmat, wheelchairs and cars, the learned part detectors commonly represent the important parts of these categories. Note that we are not given any part localization information in training, our approach can automatically learn the discriminative parts in these categories. Figure 4(b) shows total response maps of part detectors. It shows that the learned part detectors have large responses to the salient regions which are discriminative for the image category, and have low responses to the cluttered backgrounds.This indicates that our algorithm can effectively derive a set of discriminative part detectors and discard the unimportant ones.

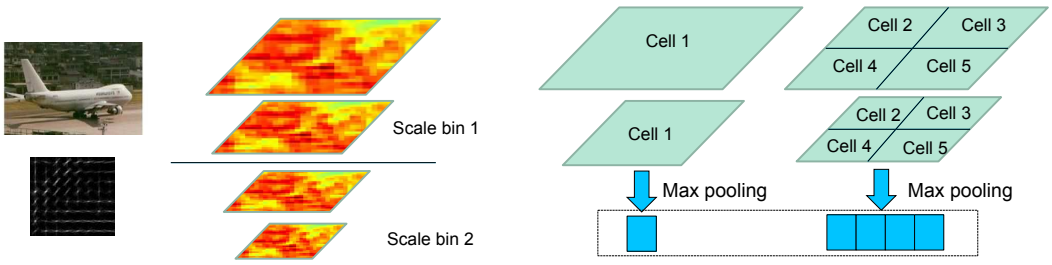
6 APPLICATIONS

Discriminative part detectors provide a mid-level and discriminative representation for an image category. We now apply them to image classification and image cosegmentation.

6.1 Image Classification

Given an image database, we learn class-specific part detectors for each category using one-vs-all training. We denote all the learned part detectors from different categories as $\Gamma = \{\Gamma_k\}_{k=1}^K$, K is the total number of part detectors. Based on our learning method for part detectors, an image I can be naturally encoded by a vector of codes $\{c_k\}_{k=1}^K$, and each code $c_k = [\max_{z \in \Omega_I} \beta_k^T \Phi(I, z) - \tau_k]_+$, corresponding to the max-pooling over the responses of part detector Γ_k to all the image parts in I.

Following object-bank [32], we improve the above coding method in a multi-scale scheme by the following steps. We resize the image resolution in 13 scaling factors ($\{2^{-2}, 2^{-1.75}, \cdots, 2^{0.75}, 2\}$) to



(a) Multi-scale image responses to a part detector.

(b) feature max-pooling (per scale bin).

Fig. 5. An illustrative example of feature max-pooling for image coding. Given an image, we compute its multi-scale response maps to each part filter. We discretize the scales uniformly into scale bins as in (a). As shown in (b), for each scale bin, we perform max-pooling of the response values over spatial cells to produce a code. The final image code is the concatenation of these codes computed for all scale bins and part detectors.

capture image parts in different scales. Then we uniformly quantitize these scales into S bins. In each scale bin, we use spatial pyramid matching (SPM) [18] by dividing the image region into spatial cells in three levels ($1 \times 1, 2 \times 2, 4 \times 4$). The response values in each spatial cell are max-pooled to produce the image code for each part detector (please refer to Figure 5 for an illustrative example). Finally, the image I is coded by concatenating all the codes computed over all part detectors and scale bins. This coding method will produce a feature vector with the length of SMK, where M is the number of cells in spatial pyramid. Given the image codes, we use a linear SVM classifier to produce the classification results.

6.2 Image Cosegmentation

For cosegmentation, we aim to segment the common objects in an image set with the same category label. Given an image set $\{I_n\}_{n=1}^N$ from the same category, we first learn discriminative part detectors $\Gamma = \{\Gamma_k\}_{k=1}^K$ from a training set with the input images as positive examples and a set of diverse background images as negative examples. As shown in Figure 4(b) and Figure 6(b), the discriminative part detectors response more strongly and frequently to the common objects of the image set, which provides a high-level common object cue for cosegmentation.

For each image I in the image set, we aim to assign labels $X = \{\boldsymbol{x}_i\}$ to pixels with $\boldsymbol{x}_i = 1$ for a foreground pixel and $\boldsymbol{x}_i = 0$ for a background pixel. This can be considered as a weakly supervised clustering problem. In particular, discriminative clustering has achieved state-of-the-art performance on cosegmentation [33], [26]. In this work, we design a novel cosegmentation algorithm by embedding the object cue provided by part detectors into the discriminative clustering framework.

Fig. 6. Cosegmentation example (the image comes from "sign" category of MSRC database). (b) Total response map. (c) Initial segmentation mask. (d) Final segmentation boundary.

We denote image feature as v_i for pixel i, and $\Psi(v_i)$ is a mapping of v_i into a high-dimensional Hilbert space $\mathcal{F}$. Discriminative clustering [33] tries to jointly infer the segment labels X and non-linear separating surface $f \in \mathcal{F}$ based on kernel SVM by minimizing:

$$E_c(X, f, d|I) = \frac{1}{|\Omega_I|} \sum_{i \in \Omega_I} [1 - \boldsymbol{x}_i(f^T \Psi(v_i) + d)]_+ + \alpha_c ||f||^2, \tag{18}$$

where d is bias term, and α_c is regularization parameter.

Discriminative clustering is an unsupervised method for cosegmentation. In our approach, we incorporate the object cue provided by part detectors and label smoothness into the above formulation. The corresponding optimization problem is:

$$\begin{aligned} \min_{X,f,d} E(X, f, d) = E_c(X, f, d|I) + \frac{1}{|\Omega_I|} \sum_{i \in \Omega_I} [E_o(\boldsymbol{x}_i|\Gamma, I) \\ + \alpha_s \sum_{j \in N(i)} E_s(\boldsymbol{x}_i, \boldsymbol{x}_j|I)], \end{aligned} \tag{19}$$

where $N(i)$ is the neighborhood of i. The above cost function is defined for an image I in the given image set, and E_o is defined based on the common object cue shared by the image set:

$$E_o(\boldsymbol{x}_i|\Gamma, I) = \begin{cases} R_i(\Gamma_k, I) - \zeta & \text{if } \boldsymbol{x}_i = 0 \\ 0 & \text{if } \boldsymbol{x}_i = 1, \end{cases} \tag{20}$$

where R_i is the value of response map in Eq.(17) at pixel i. Obviously, this model prefers to assign a foreground label to a pixel with $\sum_k R_i(\Gamma_k, I) > \zeta$, and ζ is automatically set for each image by enforcing that pixels above this threshold occupy at most 40% of the image area. E_s is a smoothness term defined as $E_s(\boldsymbol{x}_i, \boldsymbol{x}_j|I) = |\boldsymbol{x}_i - \boldsymbol{x}_j| \exp(-\frac{||v_i^c - v_j^c||_2^2}{2\sigma})$ as in [34], where v_i^c is color vector at pixel i, and σ is the mean of the squared distances between adjacent colors over the image. E_s is submodular and encourages the segmentation boundary to align with strong edges.

We optimize Eq.(19) by alternatively inferring the SVM parameters $\{f, d\}$ and the segmentation label X. Given X, $\{f, d\}$ can be found by minimizing E_c since it is the only term that depends on f and d

in Eq.(19). This can be done by a standard kernel SVM algorithm. Given $\{f, d\}$, the segmentation label X can be computed by minimizing Eq.(19) with fixed f, d, which can be efficiently optimized by graph cuts [35]. We initialize X by solving:

$$\text{argmin}_X \{ \sum_{i \in \Omega_I} [E_o(\boldsymbol{x}_i|\Gamma, I) + \alpha_s \sum_{j \in N(i)} E_s(\boldsymbol{x}_i, \boldsymbol{x}_j|I)] \}, \tag{21}$$

which is based on the object cue and label smoothness.

In our implementation, the feature vector v is the concatenation of HOG features v^h and color features v^c with length L_h and L_c respectively. Color values are scaled to [0, 1]. In kernel SVM, we use the kernel $K(v_i, v_j) = \exp(-\lambda_c(\frac{1}{L_h}||v_i^h - v_j^h||_2^2 + \frac{1}{L_c}||v_i^c - v_j^c||_2^2))$ with $\lambda_c = 5$. It is a valid kernel since it is the product of two radial basis kernels.

In implementation, we model the image cosegmentation problem at the superpixel level instead of the pixel level. An image is divided into non-overlapping superpixels produced by the efficient algorithm in [36]. Then, for each image, we define a graph whose nodes are the superpixels and edges correspond to adjacency relationship between superpixels. Based on this graph, the costs E_c, E_o in loss function of Eq.(19) are defined over superpixels. The superpixel level features (HOG, color and part response values) for each superpixel are the average of these pixel level features over all pixels within each superpixel.

Figure 6 illustrates an example of the cosegmentation procedure for an image set containing the "sign". Assume that we already learned a set of discriminative part detectors for the given cosegmentation image set, we first compute the total response map of an image to these part detectors (Fig. 6(b)), then produce the initial segmentation result by optimizing Eq.(21) (Fig. 6(c)). Starting from this initial segmentation, we iteratively optimize Eq.(19) to produce the final cosegmentation result (Fig. 6(d)).

7 EXPERIMENTS

In this section, we first present some implementation details, then present qualitative results for image classification and cosegmentation. The source codes for part learning and the applications to classification and cosegmentation are published online (*https://github.com/exploreman/discriminative_parts*).

7.1 Experimental setting

To learn part detectors, we extract dense HOG features at eight-pixel intervals, and each image part is represented as the concatenation of all HOG features in the corresponding region. The discriminative part detectors are learned in one-vs-all mode for each dataset. When training the part detectors,, we utilize multiple part templates sizes (8×8, 6×6, 4×4 feature cells) to capture features at different scales. 1000 part detectors are initialized for each category. The regularization parameter λ controls the sparsity of the solution. We have fixed it to 0.005 in all experiments, which retains about 10-15% of the

part detectors after optimization. Please see sections 7.2.1 and 7.2.2 for an investigation of the effect of λ and part initialization on the classification performance.

In the conference version of this work [16], we learn the part detectors based on the training images in their original resolution when we optimize Eq.(7) using Algorithm 1. We now have re-implemented Algorithm 1 based on training images in a multi-scale pyramid. In this setting, each training image is represented by a pyramid in thirteen successive scales ($\{2^{-2}, 2^{-1.75}, \cdots, 2^{0.75}, 2\}$), and HOG features are extracted from the image pyramid in each scale. The part detectors are then learned with the training images in HOG pyramids using Algorithm 1. As shown in the following paragraphs, this multi-scale implementation consistently produces significantly improved results for both image classification and cosegmentation.

7.2 Experiments on Image Classification

We test our classification method on four representative image databases for scene categorization (15-Scenes [18], MIT-indoor [37]), object recognition (Caltech-101 [38]), and event categorization (UIUC-Sports [39]). We use mean accuracy (i.e., the average of per-class accuracies in a database) to measure classification performance. In all the experiments, "Ours_singleScale" denotes the results produced by our previous implementation [16], and "Ours_multiScale" denotes the results produced by our current multi-scale version.

TABLE 1

Comparison on 15-Scenes database.

Single feature		*Multiple features*	
Methods	Accuracy	Methods	Accuracy
Sparse-coding [4]	80.3 ± 0.9	Object-bank [32]	80.9
SPM [18]	81.4 ± 0.5	BSPR [40]	88.9 ± 0.6
Graph-matching [41]	82.1 ± 1.1	Su et al. [42]	87.8 ± 0.5
DSS [43]	85.5 ± 0.6	Xiao et al. [44]	88.1
LPR [45]	85.8	Hybrid-Parts + Gist-color+SP [46]	86.30
ISPR [47]	85.08 ± 0.01	ISPR + FV [47]	$\mathbf{91.6 \pm 0.05}$
Hybrid-Parts [46]	84.7		
MIDL [48]	86.35		
Ours_singleScale [16]	86.0 ± 0.8		
Ours_multiScale	$\mathbf{87.2 \pm 0.5}$		

15-Scenes. This database [18] is composed of 15 categories of indoor and outdoor scenes with 4485 images. We use 10 splits of train / test data to measure the mean and standard deviation of accuracies across different categories. In each split, 100 random images are taken as training images for each category and all the other images are taken as test images. Table 1 shows comparison results on 15-Scenes by different algorithms. Our discriminative part detectors perform significantly better than the

low-level visual words in [18], [4] and high-level object detectors in [32]. Our algorithm performs better than all the algorithms using a single type of feature. The highest result on this database is 91.6% in [47] which combines ISPR features and Fisher vector (FV) [5] features. Using a single type of ISPR feature, this approach achieves mean accuracy of 85.08% which is lower than ours 87.2% using HOG feature. Obviously, our method can potentially be improved by combining several types of features, but this is not the focus of this work.

TABLE 2

Comparison on MIT-indoor 67 scenes categorization.

Methods	Accuracy
DPM [49]	30.4
DPM + GIST + SPM [49]	43.1
Object-bank [32]	37.6
DiscPatches [11]	38.1
LPR-LIN [45]	44.8
Hybrid-parts [46]	39.8
Hybrid-parts + GIST + SPM [46]	47.2
BoP [50]	43.55
MIDL [48]	50.15
Mode Seeking [12]	**64.03**
Ours_singleScale [16]	51.4
Ours_multiScale	58.1

MIT-indoor. This database contains 15620 images belonging to 67 categories of indoor scenes. It is a challenging database for categorizing indoor scenes because of the large ambiguities between categories. We use the same split of train / test data as in [37], and around 80 images are selected for training, and 20 images for testing for each category. Table 2 shows a comparison of our method with state-of-the-art algorithms on this database. We learn a total of 6372 (9.5% of the number of initial detectors) part detectors for 67 classes, and achieve 58.1% in mean accuracy using a single type of HOG features. Compared to related mid-level feature learning algorithms, our part detectors perform significantly better than discriminative patches learned by discriminative clustering [11], bag of parts model in [50], multiple instance dictionary learning approach in [48]. Though we achieve lower mean accuracy than the mode seeking algorithm [12], our result is produced by 6372 part detectors, which is much less than 13400 elements in [12]. Moreover, compared to the visual element discovery [12] and bag-of-parts models [50], our approach learns and selects discriminative parts in a more principled way by simply optimizing a latent-SVM model.

Caltech101. This database [38] contains 101 categories of objects and 40 to 800 images per category. We randomly split the database into train / test set and each category has 30 images for training. Table 3 compares the results of our approach with the other algorithms. Our learned discriminative part

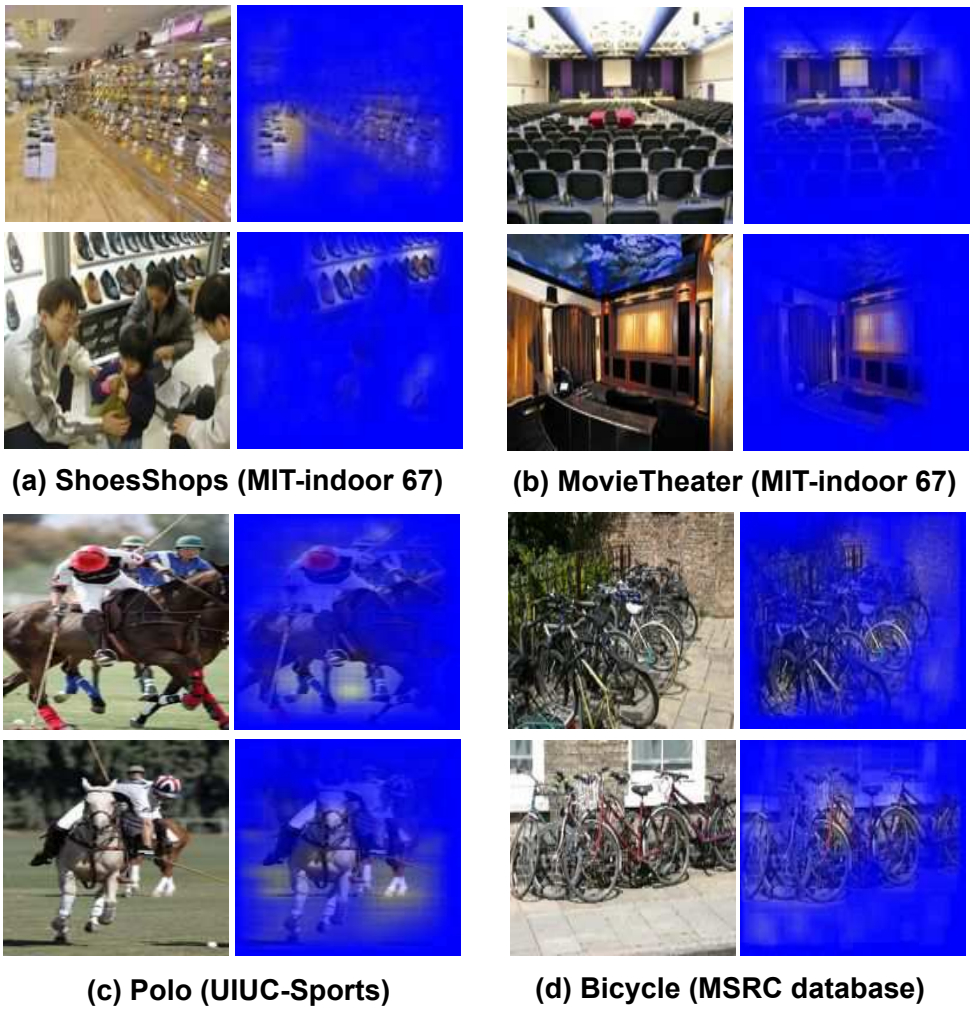


Fig. 7. More examples of the total response maps of images to the learned class-specific part detectors. (*Best viewed in color.*)

detectors achieve a competitive result of 81.6% mean accuracy on this database using a single type of HOG feature[1]. Graph matching [41] performs comparable to ours (80.3% vs. ours 81.6%) on Caltech-101 using a kernel method defined by dense matching. However, it achieves significantly lower results on 15-Scenes as shown in Table 1, probably because objects in Caltech-101 are well aligned and can be densely matched with higher accuracy.

UIUC-Sports. This database [39] contains eight categories of sport events, e.g., rowing, badminton, polo, rock climbing, etc. Following [39], we randomly take 70 images per category for training and the remaining data for testing in 10 rounds. Table 4 shows comparison results on this database. Our algorithm

1. The state-of-the-art result on Caltech-101 using multiple features is 84.3%, achieved in [51] by multiple kernel learning.

TABLE 3
Comparison on Caltech-101 database using a single feature.

Methods	Accuracy
SPM [18]	64.4 ± 0.8
Macro-feature [52]	75.7 ± 1.1
Sparse-coding [4]	73.2 ± 0.5
Multi-way pooling[53]	77.1 ± 0.7
Graph-matching[41]	80.3 ± 1.2
Ours_singleScale [16]	78.8 ± 0.5
Ours_multiScale	$\mathbf{81.6 \pm 0.6}$

achieves significantly higher results than the hybrid-parts [46], object bank [32], sparse coding [4], LPR [45] and LSA [54]. Though our result is lower than the multiple instance dictionary learning (MIDL) algorithm [48] on this database, our results are significantly higher than MIDL on the other two databases of MIT-indoor and 15-Scenes as shown in Tables 1 and 2.

TABLE 4
Comparison on UIUC-Sports database.

Methods	Accuracy
Hybrid-parts [46]	84.5
Object-bank [32]	76.3
Sparse-coding [4]	82.7 ± 1.74
LPR [45]	86.25
LSA[54]	82.3 ± 1.84
MIDL[48]	$\mathbf{88.47 \pm 2.32}$
Ours_singleScale [16]	86.4 ± 0.88
Ours_multiScale	86.8 ± 0.95

In summary, our learned discriminative part detectors perform quite competitive compared to the state-of-the-art algorithms on standard benchmarks. Moreover, using image pyramids to learn the part detectors consistently improves results compared to our previous implementation [16]. Figure 7 shows examples of the total response maps of the learned category-specific part detectors to the images sets. It shows that the learned part detectors response well to the discriminative regions in each category, and the non-informative background clutters are removed.

7.2.1 Effect of regularization parameter λ on performance

The regualization parameter λ in Eq.(7) determines the degree of sparsity imposed on the part detectors. Theoretically, increasing λ imposes higher sparsity on the part detectors, i.e., the selection of fewer number of part detectors with non-zero part templates. Figure 8 shows the effect of different λ values on

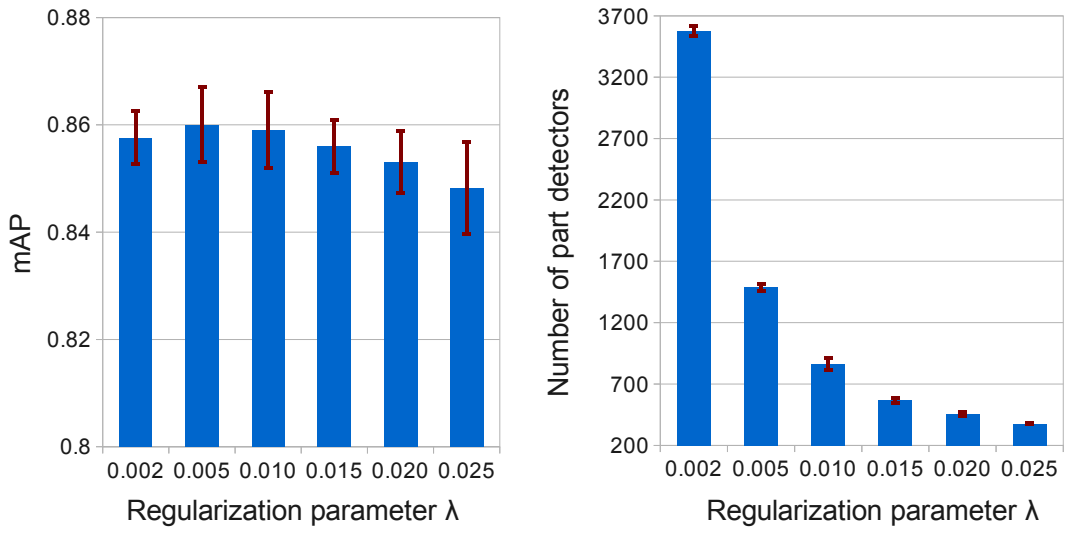


Fig. 8. The effect of the regularization parameter on classification performance for the 15-Scenes database.

the performance tested on 15-Scenes database. With the increase of λ, we observe that the classification accuracy increases then decreases. However, it is quite stable to the exact value of λ in the interval $[0.002, 0.015]$. On the other hand, with the increase of λ, the number of selected part detectors decreases fast as shown in the right subfigure in Figure 8. We achieve a competitive result of 84.8% with only 378 part detectors (much fewer than the number of words in BoWs model [18]) with $\lambda = 0.025$.

7.2.2 Effect of part initialization on performance

In the above experiments, we initialize the part detectors using the patch clustering method in Section 3.1. Now we compare classification performance using random initialization instead of patch clustering. In both cases, we initialize 1000 initial part detectors for each category. Using random initialization (i.e., randomly cropping image parts from positive training images as the initial part templates), the final learned part detectors produced 85.8% mean accuracy on the 15-Scenes database, which is lower than 87.2% using patch clustering for initialization. This is reasonable, because the cluster centers of image parts in positive training images can represent the positive images in a more compact and complete way than the randomly cropped positive patches.

We have also tested the effect of the number of initial part detectors on the final classification results. Using patch clustering for part initialization, we learned and tested the part detectors for image classification with 300, 900, 1500, 2100, 2700 initialized part detectors for each category on 15-Scenes database. With the same split of train / test data, we produced 86.7% 86.5%, 87.1%, 86.5% and 86.6%

in mean accuracy respectively. This shows that classification performance is stable with respect to the number of initial part detectors.

7.3 Experiments on Image Cosegmentation

As discussed in Section 6.2, the total response maps of category-specific part detectors provide common object cues for images containing the same objects and diverse backgrounds. We test our cosegmentation algorithm on the MSRC database which is a commonly used database for testing binary cosegmentation algorithms [33], [23], [24]. For each object category, we take the corresponding image set as positive training data, and all the other images in the dataset (they do not contain the same object) as negative training data. Then we learn object-specific part detectors, and obtain the final cosegmentation results using the algorithm proposed in Section 6.2. The parameters of cosegmentation model in Eq. (19) are set to $\alpha_c = 1, \alpha_s = 0.25$. We utilize the intersection-over-union score [26] to measure segmentation accuracy.

Table 5 shows comparison results between our algorithm and the state-of-the-art cosegmentation algorithms. The algorithm of [24] fails to converge on four classes. As shown in the table, our initial segmentation based on object cues alone already achieves better results than the method in [23]. Our full algorithm achieves the highest accuracy on this database. As before, the part detectors learned using an image pyramid produce better results (denoted as ours_Multiscale) than these learned from the single-scale training images (denoted as ours_Singlescale [16]). Figure 9 shows examples of cosegmentation results.

8 Conclusion

In this work, we have proposed a novel latent SVMs with group sparsity to learn discriminative part detectors for image recognition. Given image-level category labels, we have shown that our model is able to learn a small number of discriminative part detectors that best discriminate the image category from the background. Contrary to related algorithms, e.g., discriminative patches or bag-of-parts models, our approach is able to optimize and select the part detectors simultaneously in an efficient and principled way by optimizing the proposed learning model. We have experimentally demonstrated that our learned model achieves state-of-the-art results for image classification and cosegmentation.

In the future, we are interested in how to incorporate the spatial or geometric information among part detectors in a graph structure for object localization and recognition. Second, we will investigate using these learned mid-level part detectors for fine-grained recognition or attributes recognition.

Acknowledgement

This work was supported by the European Research Council (VideoWorld project). Jian Sun was also supported by the 973 program (2013CB329404), NSFC projects (61472313, 11131006) and NCET-12-

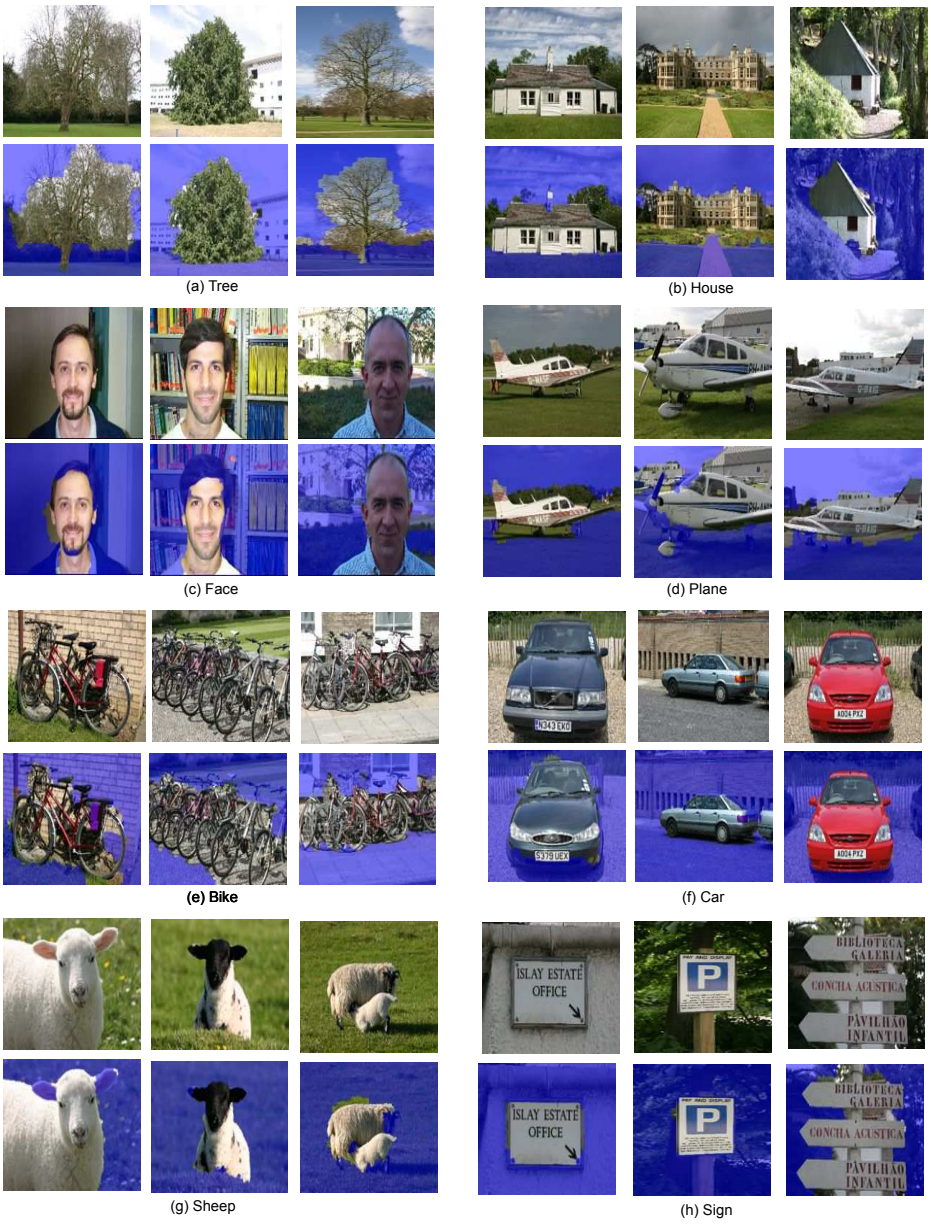


Fig. 9. Cosegmentation results on categories of “Tree”, “House”, “Face”, “Plane”, “Bike”,“Car”, “Sheep”, “Sign” in MSRC database.

TABLE 5

Comparison of the proposed cosegmentation method with Joulin et al. [33], [26], Kim et al. [23], and Mukherjee et al. [24]. "Init_singleScale" and "Init_multiScale" indicates the initial segmentation of our approach with training in modes of "singleScale" and "multiScale".

Datasets	Images	[33]	[26]	[23]	[24]	Init_singleScale [16]	Ours_singleScale [16]	Init_multiScale	Ours_multiScale
Bike	30	42.3	43.3	29.9	42.8	46.5	**50.7**	46.6	48.1
Bird	30	33.2	**47.7**	29.9	–	22.8	31.0	21.8	**32.4**
Car	30	59.0	59.7	37.1	52.5	55.0	**61.5**	57.2	59.2
Cat	24	30.1	31.9	24.4	5.6	36.5	48.0	42.1	**49.7**
Chair	30	37.6	39.6	28.7	39.4	39.4	48.9	40.5	**49.9**
Cow	30	45.0	**52.7**	33.5	26.1	38.2	45.6	43.7	**54.6**
Dog	26	41.3	41.8	33.0	–	32.4	**46.6**	33.5	44.6
Face	30	**66.2**	70.0	33.2	40.8	48.4	**50.3**	46.6	47.6
Flower	30	50.9	51.9	40.2	–	50.2	**75.7**	51.0	69.5
House	30	50.5	51.0	32.2	**66.4**	51.1	61.5	52.1	**62.7**
Plane	30	21.7	21.6	25.1	**33.4**	28.2	28.1	33.7	**39.8**
Sheep	30	60.4	66.3	60.8	45.7	47.8	**65.2**	45.9	62.8
Sign	30	55.2	58.9	43.2	–	50.9	69.9	58.0	**73.8**
Tree	30	60.0	67.0	61.2	55.9	55.8	**70.1**	54.2	66.7
Average		46.7	50.2	36.6	–	43.1	53.8	44.8	**54.4**

0442.

REFERENCES

[1] D. G. Lowe, "Object recognition from local scale-invariant features," in *CVPR*, 1999.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.

[3] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *ECCV workshop on statistical learning in computer vision*, 2004.

[4] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *CVPR*, 2009.

[5] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *International journal of computer vision*, vol. 105, no. 3, pp. 222–245, 2013.

[6] P. Arbeláez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik, "Semantic segmentation using regions and parts," in *CVPR*, 2012.

[7] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3d human pose annotations," in *ICCV*, 2009.

[8] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. Efros, "What makes paris look like paris?" *ACM TOG*, vol. 31, no. 4, p. 101, 2012.

[9] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE T. PAMI*, vol. 32, no. 9, pp. 1627–1645, 2010.

[10] M.Juneja, A. Vedaldi, C. V. Jawahar, and A. Zisserman, "Blocks that shout: Distinctive parts for scene classification," in *CVPR*, 2013.

[11] S. Singh, A. Gupta, and A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *ECCV*, 2012.

[12] C. Doersch, A. Gupta, and A. A. Efros, "Mid-level visual element discovery as discriminative mode seeking," in *Advances in Neural Information Processing Systems*, 2013, pp. 494–502.

[13] B. Yao, X. Jiang, A. Khosla, A. Lin, L. Guibas, and L. Fei-Fei, "Human action recognition by learning bases of action attributes and parts," in *ICCV*, 2011.

[14] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 689–696.

[15] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *Image Processing, IEEE Transactions on*, vol. 15, no. 12, pp. 3736–3745, 2006.

[16] J. Sun and J. Ponce, "Learning discriminative part detectors for image classification and cosegmentation," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3400–3407.

[17] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *CVPR*, 2010.

[18] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006.

[19] H. Azizpour and I. Laptev, "Object detection using strongly-supervised deformable part models," in *ECCV*, 2012.

[20] Z. Jiang, Z. Lin, and L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent k-svd," in *CVPR*, 2011.

[21] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Discriminative learned dictionaries for local image analysis," in *CVPR*, 2008.

[22] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?" *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[23] G. Kim, E. P. Xing, L. Fei-Fei, and T. Kanade, "Distributed cosegmentation via submodular optimization on anisotropic diffusion," in *ICCV*, 2011.

[24] L. Mukherjee, V. Singh, and J. Peng, "Scale invariant cosegmentation for image groups," in *CVPR*, 2011.

[25] S. Vicente, C. Rother, and V. Kolmogorov, "Object cosegmentation," in *CVPR*, 2011.

[26] A. Joulin, F. Bach, and J. Ponce, "Multi-class cosegmentation," in *CVPR*, 2012.

[27] G. Kim and E. P. Xing, "On multiple foreground cosegmentation," in *CVPR*, 2012.

[28] Y. Chai, E. Rahtu, V. Lempitsky, L. Van Gool, and A. Zisserman, "Tricos: A tri-level class-discriminative co-segmentation method for image classification," in *ECCV*, 2012.

[29] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B*, vol. 68, no. 1, pp. 49–67, 2005.

[30] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[31] J. Duchi and Y. Singer, "Efficient learning using forward-backward splitting," in *NIPS*, 2009.

[32] L. Li, H. Su, E. Xing, and L. Fei-Fei, "Object bank: A high-level image representation for scene classification and semantic feature sparsification," in *NIPS*, 2010.

[33] A. Joulin, F. Bach, and J. Ponce, "Discriminative clustering for image co-segmentation," in *CVPR*, 2010.

[34] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *ACM TOG*, vol. 23, no. 3, pp. 309–314, 2004.

[35] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE T. PAMI*, vol. 26, no. 9, pp. 1124–1137, 2004.

[36] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE T. PAMI*, vol. 34, no. 11, pp. 2274–2282, 2012.

[37] A. Quattoni and A.Torralba, "Recognizing indoor scenes," in *CVPR*, 2009.

[38] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *CVPR workshop on generative-model based vision*, 2004.

[39] L.-J. Li and L. Fei-Fei, "What, where and who? classifying events by scene and object recognition," in *ICCV*, 2007.

[40] S. Yan, X. Xu, D. Xu, S. Lin, and X. Li, "Beyond spatial pyramids: A new feature extraction framework with dense spatial sampling for image classification," in *ECCV*, 2012.

[41] O. Duchenne, A. Joulin, and J. Ponce, "A graph-matching kernel for object categorization," in *ICCV*, 2011.

[42] Y. Su and F. Jurie, "Visual word disambiguation by semantic contexts," in *ICCV*, 2011.

[43] G. Sharma, F. Jurie, and C. Schmid, "Discriminative spatial saliency for image classification," in *CVPR*, 2012.

[44] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *CVPR*, 2010.

[45] F. Sadeghi and M. F. Tappen, "Latent pyramidal regions for recognizing scenes," in *ECCV*, 2012.

[46] Y. Zheng, Y.-G. Jiang, and X. Xue, "Learning hybrid part filters for scene recognition," in *ECCV*, 2012.

[47] D. Lin, C. Lu, R. Liao, and J. Jia, "Learning important spatial pooling regions for scene classification," in *CVPR*, 2014.

[48] X. Wang, B. Wang, X. Bai, W. Liu, and Z. Tu, "Max-margin multiple-instance dictionary learning," in *Proceedings of The 30th International Conference on Machine Learning*, 2013, pp. 846–854.

[49] M. Pandey and S. Lazebnik, "Scene recognition and weakly supervised object localization with deformable part-based models," in *ICCV*, 2011.

[50] M. Juneja, A. Vedaldi, C. Jawahar, and A. Zisserman, "Blocks that shout: Distinctive parts for scene classification," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 923–930.

[51] J. Yang, Y. Li, Y. Tian, L. Duan, and W. Gao, "Group-sensitive multiple kernel learning for object categorization," in *CVPR*, 2009.

[52] Y. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *CVPR*, 2010.

[53] Y. Boureau, N. Le Roux, F. Bach, J. Ponce, and Y. LeCun, "Ask the locals: multi-way local pooling for image recognition," in *ICCV*, 2011.

[54] L. Liu, L. Wang, and X. Liu, "In defense of soft-assignment coding," in *ICCV*, 2011.

Jian Sun received the B.S. degree from the University of Electronic Science and Technology of China in 2003 and the Ph.D. degree in applied mathematics from Xian Jiaotong University in 2009. He worked as a visiting student in Microsoft Research Asia from November 2005 to March 2008, a postdoctoral researcher in University of Central Florida from August 2009 to April 2010, and a postdoctoral researcher in willow project team of Ecole Normale Supérieure de Paris and INRIA from Sept. 2012 to August 2014. He now serves as an associate professor in the school of mathematics and statistics of Xian Jiaotong University. His current research interests are image categorization, object detection and image processing (e.g., image deblurring and super-resolution).

Jean Ponce is a computer science professor at Ecole Normale Supérieure (ENS) in Paris, France, where he heads the ENS/INRIA/CNRS Project-team WILLOW. Before joining ENS, he spent most of his career in the US, with positions at MIT, Stanford, and the University of Illinois at Urbana-Champaign, where he was a full professor until 2005. Jean Ponce is the author of over 120 technical publications in computer vision and robotics, including the textbook Computer Vision: A Modern Approach. He is an IEEE Fellow, served as editor-in-chief for the International Journal of Computer Vision from 2003 to 2008, and chaired the IEEE Conference on Computer Vision and Pattern Recognition in 1997 and 2000, and the European Conference on Computer Vision in 2008.